

**REMARKS**

The Examiner is thanked for the performance of a new search, for considering the references submitted by the Applicants in the Information Disclosure Statements filed on March 28, 2005 and April 14, 2005, and for entering the Request for Continuing Examination (RCE) filed by the Applicants on August 8, 2005.

No claims have been added, canceled, or amended. Hence, Claims 1-18 are pending in the application.

I. THE PRESENT OFFICE ACTION IS IMPROPERLY MADE FINAL

The present final Office Action is a first Office Action immediately subsequent to the filing of a Request for Continuing Examination on August 8, 2005 in the present application.

In section titled “VIII. FIRST ACTION FINAL AFTER FILING AN RCE”, MPEP § 706.07(h) states:

The action immediately subsequent to the filing of an RCE with a submission and fee under 37 CFR 1.114 may be made final only if the conditions set forth in MPEP § 706.07(b) for making a first action final in a continuing application are met.

MPEP § 706.07(b) states:

However, it would not be proper to make final a first Office Action in a continuing or substitute application where that application contains **material which was presented in the earlier application after final rejection or closing of prosecution but was denied entry because (A) new issues were raised that required further consideration and/or search, or (B) the issue of new matter was raised.** (Emphasis added.)

Thus, it would not be proper to make final a first Office Action after the filing of an RCE in an application that contains material presented in the application after a previous final rejection,

where the material was denied entry because new issues were raised by the material that required further consideration and/or search.

In response to the final Office Action mailed in the present application on April 6, 2005, the Applicants filed a Reply to Final Office Action on June 6, 2005. In the Reply to Final Office Action filed on June 6, 2005, the Applicants amended Claims 1 and 10 in order to place the application in condition for allowance. The Advisory Action mailed July 6, 2005 denied entry of these amendments because it considered the amended Claims 1 and 10 to include new limitations that "require at least reconsideration of the applied references and possibly a new search as well." In response to the Advisory Action, the Applicants filed the Request for Continuing Examination on August 8, 2005 along with a copy of the previously submitted Reply to Final Office Action.

As the present Office Action indicates, the RCE of August 8, 2005 and the amendments to Claims 1 and 10 originally submitted on June 6, 2005 were entered and a new search was performed. Thus, the present final Office Action is the first Office Action after the filing of an RCE, which introduced in the application amendments that were originally made after a prior final rejection but were denied entry because reconsideration and possibly a new search would be required. Under these circumstances, according to the provisions of MPEP §§ 706.07(h) 706.07(b) the present Office Action is improperly made final.

For the above reasons, it is respectfully submitted that the present Office Action is improperly made final. Reconsideration and withdrawal of the finality of the rejections in the present Office Action is respectfully requested.

## II. SUMMARY OF THE REJECTIONS

Claims 1-18 have been rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Rich et al., U.S. Patent No. 6,457,065 (“RICH”).

## III. REJECTIONS BASED ON THE CITED ART

### A. CLAIMS 1 AND 10

Claims 1 and 10 recite the following features:

**receiving a command to perform one or more file system operations;**  
in response to said command, performing a plurality of operations including  
**said one or more file system operations;**  
wherein the step of performing the plurality of operations includes:  
    **performing a first subset of said plurality of operations as part of a**  
        **first transaction; and**  
    **performing a second subset of said plurality of operations as part of a**  
        **second transaction that is nested in said first transaction,**  
    **wherein each of said one or more file system operations is included in at**  
        **least one of the first subset of said plurality of operations and the**  
        **second subset of said plurality of operations.**

The features of Claims 1 and 10 indicate that a command to perform one or more file system operations is received. In response to the command, a plurality of operations that includes the one or more file system operations is performed. Performing the plurality of operations includes: performing a first subset of the plurality of operations as part of a first transaction, where the first subset of the plurality of operations includes some of the one or more file system operations; and performing a second subset of the plurality of operations as part of a second transaction that is nested in the first transaction, where the second subset of the plurality of operations also includes some of the one or more file system operations. Thus, the features of Claims 1 and 10 clearly indicate that the one or more file system operations identified in the received command are performed as part of nested transactions.

It is respectfully submitted that the above features are not disclosed, taught, or suggested by RICH.

1. RICH does not describe, either explicitly or implicitly, that one or more file system operations are performed as part of nested transactions

In response to Applicants' prior argument that RICH does not describe the performance of file system operations, the Office Action states in page 6, numbered paragraph 16:

Rich is concerned with replication of remote objects on a persistent store, where Rich explicitly indicates that the persistent store may be a file system. Objects are first updated internally, without updating the persistent store (col. 7, lines 66 – col. 8, line 4). Then, the user decides whether or not to commit the changes, wherein the updated version is then copied to the persistent store if the update is committed (col. 8, lines 4-8). When the changes are committed to the persistent store, which may be a file system, the *must* be a file system operation to effect the changes. There is no way that the file system may be updated without some sort of a write to the file system. It is granted that Rich does not use the specific language of "file system operation"; the file system operation is inherent in the step of committing the transaction. (Italics in the original.)

Thus, the Office Action considers any operations in RICH, which may inherently be performed to copy an updated object to a permanent store that is a file system, to be equivalent to the file system operations feature of Claims 1 and 10.

RICH, however, makes a very clear distinction between operations that update objects and operations that copy updated objects to the persistent store. While RICH may be describing that operations which update objects participate in transactions, RICH does NOT teach or suggest that any operations that copy objects to the permanent store are part of any transactions.

Specifically, RICH defines "a technique for using a transactional approach to **maintain consistency in updating objects**, where the **tasks a user** (or application program) **performs are organized into transactions.**" (Col. 7, lines 63-66; emphasis added.) "[A] transaction represents a logical group of **changes to one or more objects** that will be performed in an

atomic manner." (RICH, col. 7, lines 56-58; emphasis added.) Thus, RICH expressly states that only a logical group of CHANGES made to OBJECTS by USERS is represented in a transaction. Furthermore, in col. 7, line 66 to col. 8, line 10, RICH states:

Using the transactional approach defined in this first related invention, **all changes that the user wants to make to an object within the scope of a transaction are made first to an internal copy** (called a "version") of the object, without actually updating the persistent, stored object itself. The user eventually decides whether to permanently commit the changes encompassed by the transaction, or whether to discard them. If the user chooses to commit, then the updated object in the version is copied to the persistent store. If the user chooses to discard (or "roll back") the changes, the persistent store is left unchanged. (Emphasis added.)

The above passage clearly states that in RICH only changes made to objects by users are part of any transactions. Furthermore, the above passage unambiguously states that any changes made to objects by a user within the scope of a transaction are NOT copied to permanent store but are kept in an internal copy, or "version", of the object; if the user chooses later to disregard these changes to the object, "**the persistent store is left unchanged.**"

Thus, the above passage makes it unambiguously clear that any operations that copy updated objects to the permanent store are NOT part of any transactions. A user must first COMMIT a transaction before any objects that were updated in the transaction can be copied to the permanent store. In contrast, Claims 1 and 10 recite that one more file system operations are performed as part of nested transactions.

With regards to nested transactions, in col. 11, lines 49-66, RICH states:

Within the scope of a nested transaction, a child transaction at any level may either commit its changes or roll them back. **If the child transaction commits, its versions of replicated remote objects (including any changed values therein) are merged to its parent transaction. The change does not actually update the remote object instance at the persistent store unless and until the top-level transaction is committed.** Further, the changes are not visible to any siblings of the committed child transaction, or to any nodes other than the parent

(such as child transactions that are siblings of this parent), until the changes propagate upward through the tree by commits at higher-level nodes. (This enables the manager and employee working at different workstations 403, 404, as described above, to work independently with consistent data values that do not change due to changes made concurrently by other users.) **If a child transaction rolls back, then the changes made by this child and all its children are discarded, without updating the remote object.** (Emphasis added.)

Thus, if changes made to objects by users are organized in nested (parent-child) transactions, any changes to child objects are NOT written to the persistent store UNLESS and UNTIL the top-level transaction IS COMMITTED. In other words, RICH clearly and unambiguously states that any operations that write updated objects to the persistent store are performed AFTER the top-level transaction in the nested transaction is committed. For this reason, any operations that write to persistent store CANNOT possibly be performed as part of the nested transactions described in RICH. Thus, RICH does not describe any operations that can be reasonably interpreted as corresponding to the file system operations recited in Claims 1 and 10.

In contrast, the one or more file system operations featured in Claims 1 and 10 are performed as part of nested transactions. Specifically, Claims 1 and 10 recite the feature of wherein each of said one or more file system operations is included in at least one of the first subset of said plurality of operations and the second subset of said plurality of operations.

The Office Action asserts that this feature is described in RICH in col. 8, lines 47-55, col. 9, lines 3-12, and col. 21, lines 51-55. This is incorrect.

In col. 8, lines 47-55, RICH describes that each transaction is provided with its own version of each Enterprise JavaBean (EJB) that it accesses, and each sub-transaction within a transaction may further have its own view of the EJB if that transaction or its children access the EJB. However, nothing in this passage from RICH has anything to do with operations that

write to the persistent store, let alone describe that such operations are included in nested transactions.

In col. 9, lines 3-12, RICH describes that a logical transactional model for a replicated distributed object system may form a tree of nested transactions, which may be used to manage changes to replicas resulting from one or more concurrent and/or nested transactions. While this passage from RICH may be describing trees of nested transactions, nothing in this passage describes that operations that write updated objects to persistent store are performed in such nested transactions.

Finally, in col. 21, lines 51-55, RICH states:

The discussions of **persistent store** herein are in terms of using a "database". However, it is to be understood that this is for ease of reference. **Any type of persistent store, however organized (such as a file system), may be used without deviating from the inventive concepts disclosed herein.** (Emphasis added)

While the above passage may be describing that a persistent store may be organized as a file system, nothing in this passage teaches or suggests that operations which write an object to the persistent store are performed as part of nested transactions. In fact, this passage from RICH is the ONLY place where RICH discusses file systems. It seems that the Office Action relies exclusively on this passage and this passage alone to bootstrap ALL its other assertions regarding the file system operations featured in Claims 1 and 10. It is respectfully submitted that in the context of the RICH disclosure, this passage is NOT sufficient on its own to support the assertions that the Office Action uses in rejecting Claims 1 and 10.

Perhaps realizing that the above passage from RICH is insufficient to support prior disclosure of the file system operations featured in Claims 1 and 10, the Office Action incorrectly asserts that the Applicants have admitted that file system operations may be

performed in a transactional approach. Specifically, in response to Applicants' previous arguments, the Office Action states in page 7, numbered paragraph 18, that "Applicants have admitted [that] prior art shows file system operations may be implemented in a transactional approach (Specification, pg. 2, lines 19-24)."

The Applicants respectfully submit that the passage in page 2, lines 19-24 of the specification is NOT an admission that file system operations may be implemented in a transactional approach. The mere inclusion of something in the "Background" section of the specification is not sufficient to qualify as applicant admitted prior art. An applicant must actually admit or label something as "prior art" in order for that something to qualify as applicant admitted prior art. Since in the present application the Applicants have not labeled the passage in page 2, lines 19-24 of the specification as "prior art", the mere presence of this passage in the "Background" section of the application is not sufficient to qualify that passage as applicant admitted prior art.

For all of the above reasons, it is respectfully submitted that RICH does not teach or suggest the feature of Claims 1 and 10 of one or more file system operations that are performed as part of nested transactions.

2. RICH does not describe, either explicitly or implicitly, the feature of Claims 1 and 10 of receiving a command to perform one or more file systems operations

In page 7, numbered paragraph 20, the Office Action asserts that the above feature of Claims 1 and 10 is equivalent to a user command in RICH that commits a transaction. This is incorrect.

As discussed above, the Office Action considers that any operations in RICH that copy updated objects to permanent store are equivalent to the file system operations featured in Claims 1 and 10. As also discussed above, RICH describes that a transaction represents only a logical group of CHANGES made to OBJECTS by USERS. (See RICH, col. 7, lines 56-66.) As shown above, however, the transactions in RICH do NOT include any operations that write copies of updated objects to permanent store. Specifically, the transactions in RICH that write updated objects to permanent store are not performed as part of nested transactions as featured in Claims 1 and 10.

Thus, if a command from a user to commit a transaction in RICH is equivalent to the command to perform one or more file system operations as featured in Claim 1, then because the operations in RICH that write to the permanent store are not part of any transactions, RICH cannot possibly describe one or more other the features of Claims 1 and 10, such as for example the features of

**performing a first subset of said plurality of operations as part of a first transaction; and**  
**performing a second subset of said plurality of operations as part of a second transaction that is nested in said first transaction.**

For this reason alone, a command from a user to commit a transaction in RICH is NOT equivalent to a command to perform one or more file system operations as featured in Claims 1 and 10.

Furthermore, the passages from RICH (col. 7, lines 56-59, col. 17, lines 51-57, col. 19, lines 30-33, and col. 21, lines 51-55) which the Office Action recites as allegedly showing “receiving a command to perform one or more file system operations” are not relevant to this feature of Claims 1 and 10.

Specifically, in col. 7, line 56-59, RICH defines a transaction as representing a logical group of changes made by users to one or more objects, which changes are to be performed in an atomic manner. This is simply a definition of what the term “transaction” means and has nothing to do with “receiving a command”.

In col. 17, lines 51-57, RICH describes the execution of a database query to retrieve a requested object from the database or data store in which the object resides. This passage describes accessing a persistent store that is a database; thus, any operations that may be performed as part of accessing the database are database operations and not file system operations as featured in Claims 1 and 10.

In col. 19, lines 25-33, with respect to its FIG. 13, RICH describes the process of writing a version of a changed object to the permanent store. Specifically, versions of objects that are changed are written to the permanent store, and versions of objects that are not changed are not written. However, since as discussed above any operations in RICH that write updated objects to permanent store are not part of nested transactions, any such operations cannot be equivalent to the file system operations in Claims 1 and 10. Thus, RICH fails to describe the feature of Claims 1 and 10 of “receiving a command to perform one or more file system operations”.

For the above reasons, it is respectfully submitted that RICH does not describe or suggest all features of Claims 1 and 10. Thus, RICH does not anticipate Claims 1 and 10 under 35 U.S.C. § 102(e), and reconsideration and withdrawal of the rejections are respectfully requested.

#### B. CLAIMS 8 AND 17

Dependent Claims 8 and 17 have been rejected under 35 U.S.C. § 102(e) as allegedly

anticipated by RICH.

Each of Claims 8 and 17 is dependent from one of independent Claims 1 and 10, and thus includes each and every feature of the corresponding independent claim. Therefore, Claims 8 and 17 are allowable for at least the reasons given above for Claims 1 and 10. In addition, Claims 8 and 17 recite the features of:

**the one or more file system operations include an operation on a folder; and the second subset of operations includes operations associated with one or more documents within the folder.**

It is respectfully submitted that RICH does not describe the above features of Claims 8 and 17.

The Office Action asserts that the above features of Claims 8 and 17 are described in col. 9, line 41 to col. 10, line 41. This is incorrect.

In col. 9, line 41, to col. 10, line 41, RICH states:

An example configuration of a distributed object system 400 having six nodes with four transactions spanning various ones of these six nodes according to the present invention is depicted in FIG. 4A. This example configuration accesses (and therefore replicates) objects stored in four separate repositories 450, 451, 452, and 453 (which may be considered as four additional server nodes of the distributed object system). Node\_(element 401) is a parent of a transaction 410 which has two child nodes 411, 412 in the local node. The child 411 then has a child 413 (i.e. a child transaction) in Node\_1.1, which has its own child 414 in Node\_1.1.2. The child 412 has a child 415 at Node\_1.2, where this node 415 has two children 416 and 417. Child 416 also has a child 418 at Node\_1.2.1. Thus, transaction 410 spans five of the six physical nodes of this example distributed object system 400, and is up to five levels deep (following the path between elements 410, 412, 415, 416, and 418). Transaction 420 has its root at Node\_1.1.402, and also has child node 421 at Node\_1.1 and descendant nodes 422, 423, 424 at Node\_1.1.1403. Transaction 430 of the example is located only on Node\_1.1.2404, and Transaction 440 is located only on Node\_1.2.1406. Note that while these four example transactions are shown as following the same hierarchy, this is for purposes of illustration and not of limitation. For example, Node\_1.1.1403 might be a parent to a transaction having its next lower level at Node\_1.2.405, followed by a next lower level at Node\_401.

As an example of using the present invention for transactions which span multiple nodes of an object system, suppose the distributed object system includes the Employee, Manager, and Address objects previously described with

reference to FIG. 1 and also Department objects and Division objects (where the underlying object model for a Division comprises multiple Departments and each Department has one Manager and one or more Employees). The objects for this model may be stored in one global repository, or they may be dispersed across multiple physical storage locations. Suppose a copy of the objects are stored in a global repository 450 at Node\_1401, and the departmental servers (such as Node\_1.1402 and Node\_1.2403) also have a replicated copy of the objects for a first department and a second department, respectively. The present invention may be used to manage the consistency of these replicated copies with the global repository. Further suppose a manager wishes to view and possibly update objects related to the employees in his department. The manager may be using a client workstation such as Node\_1.1.1403. The manager then, according to the present invention, replicates a copy of the Employee object (and perhaps their Address objects as well) for each of his employees from departmental server 402 to his client workstation 403. An employee in this manager's department may, at the same time, wish to view and possibly update his own employee or address information. The employee may be using a client workstation such as Node\_1.1.2404. The employee replicates a copy of his own Employee or Address object from departmental server 402 to his client workstation 404. Using the techniques of the present invention, as will be described in detail below, the integrity and consistency of any common objects that are replicated from departmental server 402 for the work being done by the manager at workstation 403 and for the work being done by the employee at workstation 404 are automatically and transparently maintained. Furthermore, the integrity and consistency of these replicated objects is automatically and transparently maintained with respect to the remote objects at global repository 401, from which the copies were replicated.

With respect to FIG. 4 of RICH, the above passage describes an example configuration of a distributed object system having six nodes with four transactions spanning various ones of these six nodes. Further, above passage gives an example of maintaining consistent updates to an object (an Employee object), which updates are initiated from users that execute applications at different nodes of RICH's distributed object system.

The Applicants cannot identify anything in the above passage that corresponds to the features of Claims 8 and 17 of a file system operation on a folder or to operations associated with documents in that folder. Furthermore, neither this passage nor any other passage in RICH teaches or suggests that any file system operations are performed on folders in a file system or

on documents within such folders. The terms “folder” or “directory”, as used in the context of a file system, are nowhere to be found in RICH; thus, RICH cannot possibly describe performing any file system operations on folders or documents within folders.

For the above reasons, it is respectfully submitted that RICH does not describe or suggest the above features of Claims 8 and 17. Thus, Claims 8 and 17 are not anticipated under 35 U.S.C. § 102(e) by RICH, and for this reason reconsideration and withdrawal of the rejections of Claims 8 and 17 are respectfully requested.

C. CLAIMS 2-7, 9, 11-16, AND 11-18

Claims 2-7, 9, 11-16, and 18 have been rejected under 35 U.S.C. § 102(e) as allegedly anticipated by RICH.

Each of Claims 2-7, 9, 11-16, and 18 is dependent from one of independent Claims 1 and 10, and thus includes each and every feature of its corresponding independent claim. Each of Claims 2-7, 9, 11-16, and 18 is therefore allowable for the reasons given above for Claims 1 and 10. In addition, each of Claims 2-7, 9, 11-16, and 18 introduces one or more additional features that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those features is not included at this time. Therefore, it is respectfully submitted that Claims 2-7, 9, 11-16, and 18 are allowable for the reasons given above with respect to Claims 1 and 10.

IV. CONCLUSION

The Applicants believe that all issues raised in the Office Action have been addressed. Further, for the reasons set forth above, the Applicants respectfully submit that all pending claims are patentable over the cited art and for this reason allowance of the pending claims is

most earnestly solicited. Reconsideration of the present application is respectfully requested in light of the amendments and remarks herein.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

To the extent necessary to make this reply timely filed, the Applicant petitions for an extension of time under 37 C.F.R. § 1.136. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,  
HICKMAN PALERMO TRUONG & BECKER LLP

Dated: October 31, 2005

Stoycho D. Draganoff  
Stoycho D. Draganoff  
Reg. No. 56,181

2055 Gateway Place, Suite 550  
San Jose, California 95110-1089  
Telephone No.: (408) 414-1080 ext. 208  
Facsimile No.: (408) 414-1076